EXHIBIT 1

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

GOOGLE LLC,

                                Plaintiff
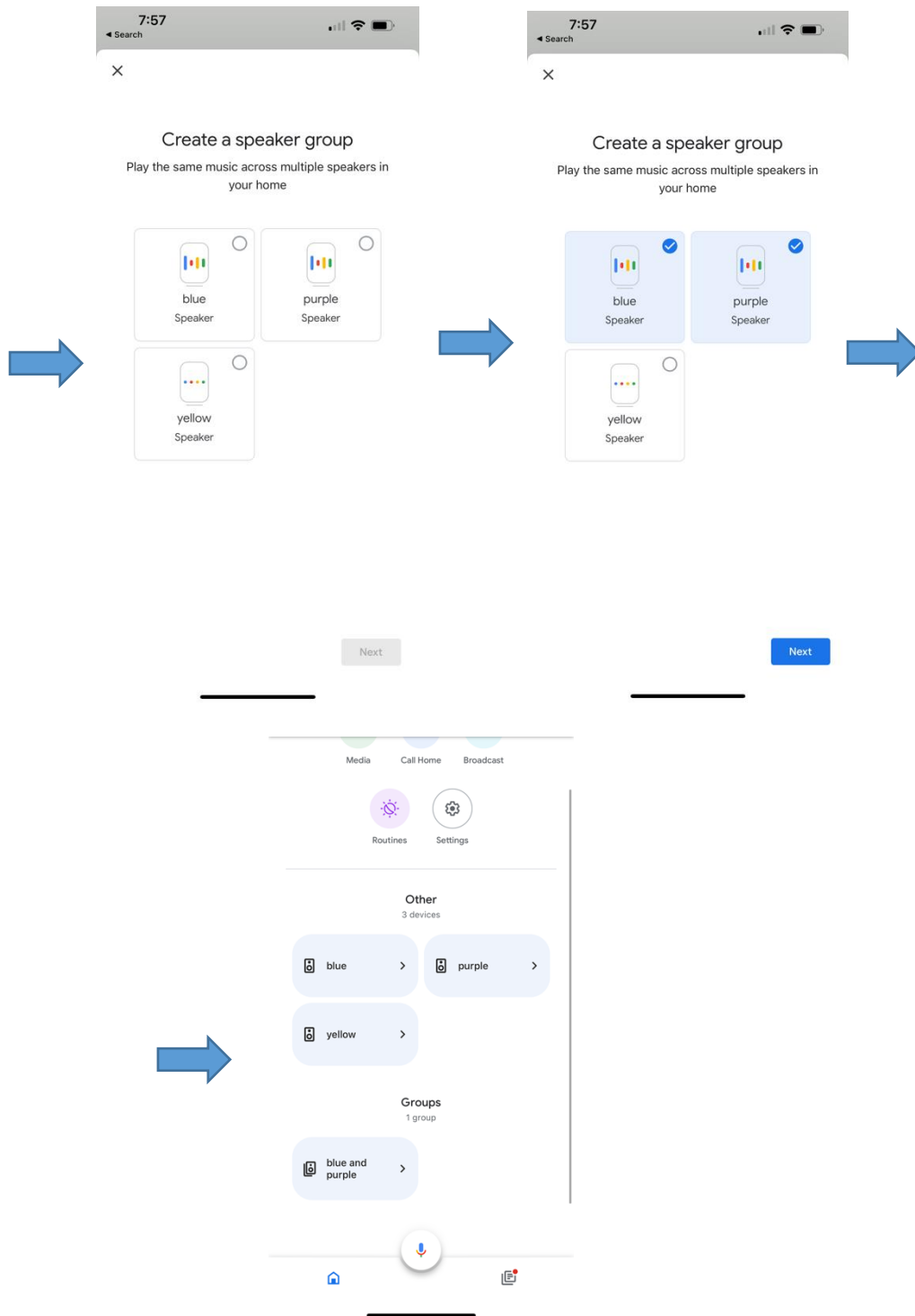
    v.

SONOS, INC.,

                                Defendant.

CASE NO. 3:20-cv-06754-WHA
Related to CASE NO. 3:21-cv-07559-WHA
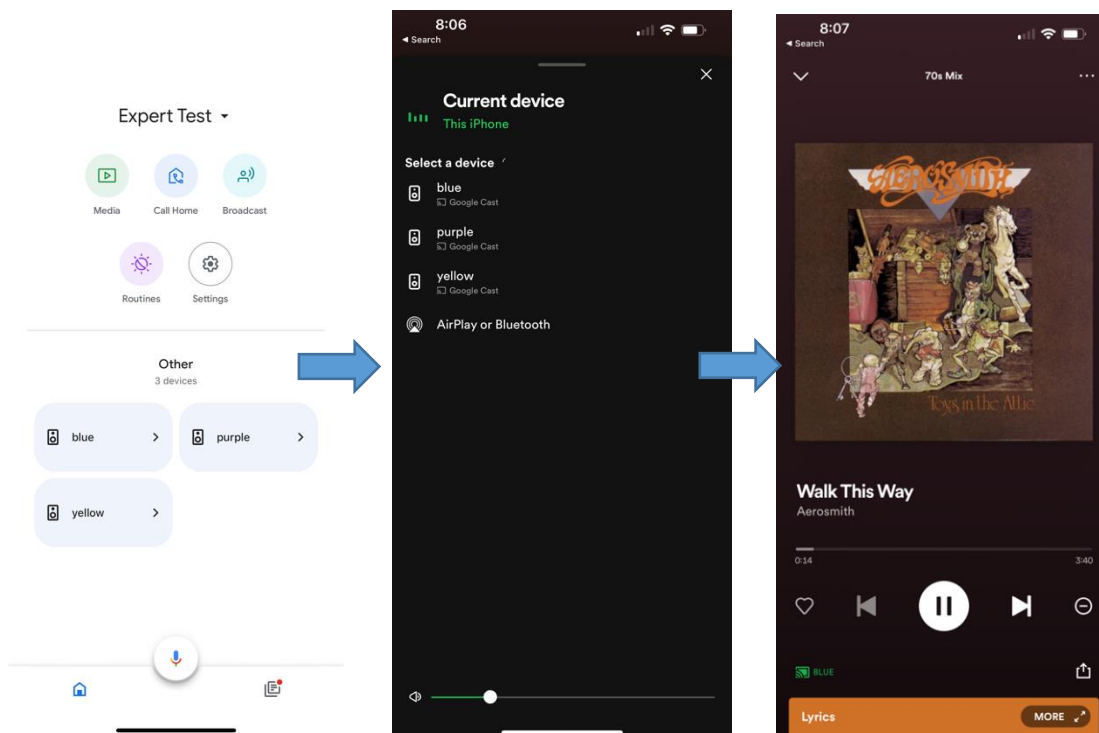
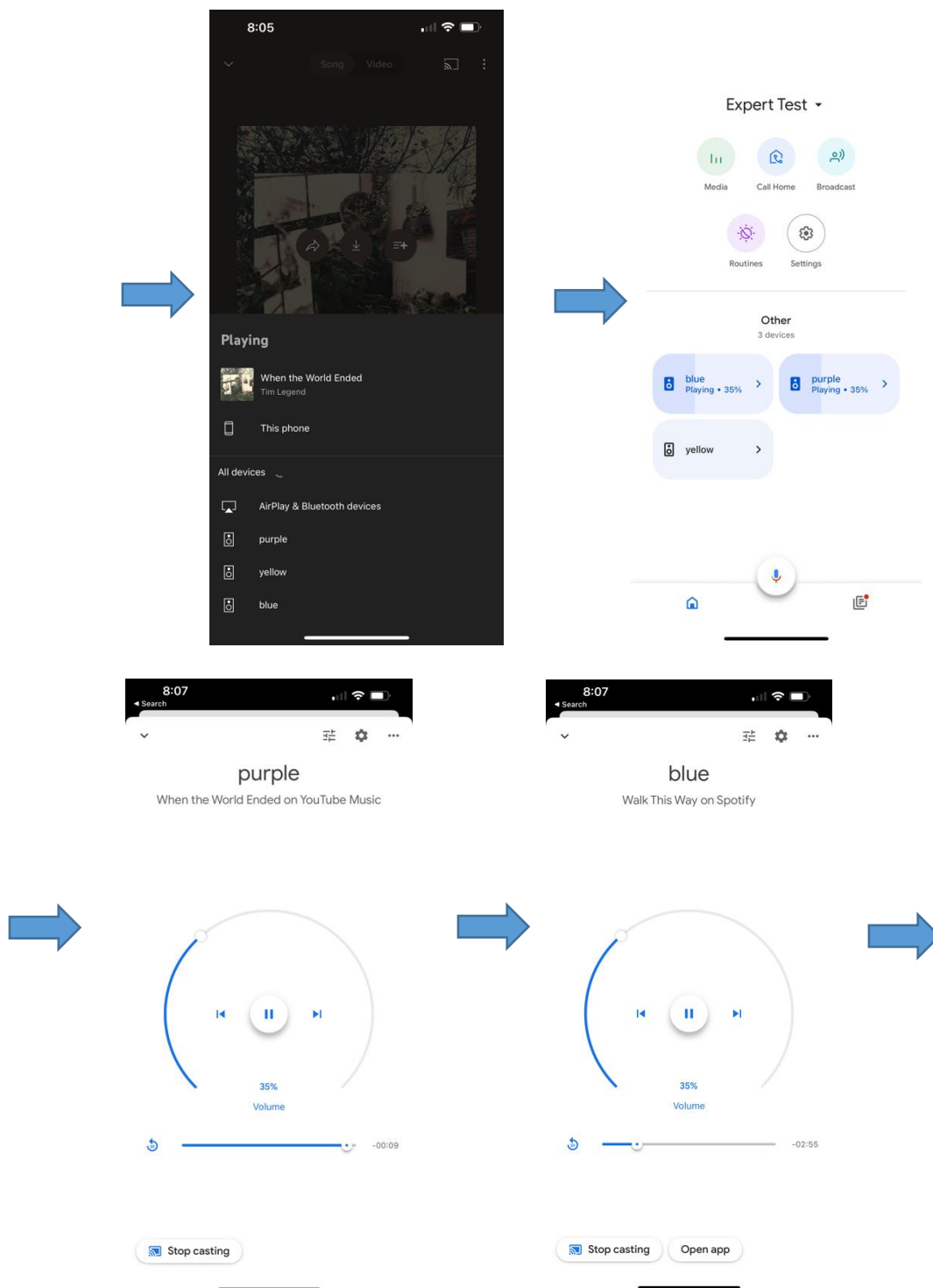# REBUTTAL EXPERT REPORT OF DAN SCHONFELD, Ph.D.

57.   Next, the scenario where two speakers playing back music are added to a group and one speaker not playing back music is illustrated below.   Music is played from Spotify to the speaker named "blue," and music from YouTube Music is played to the speaker named "purple,"
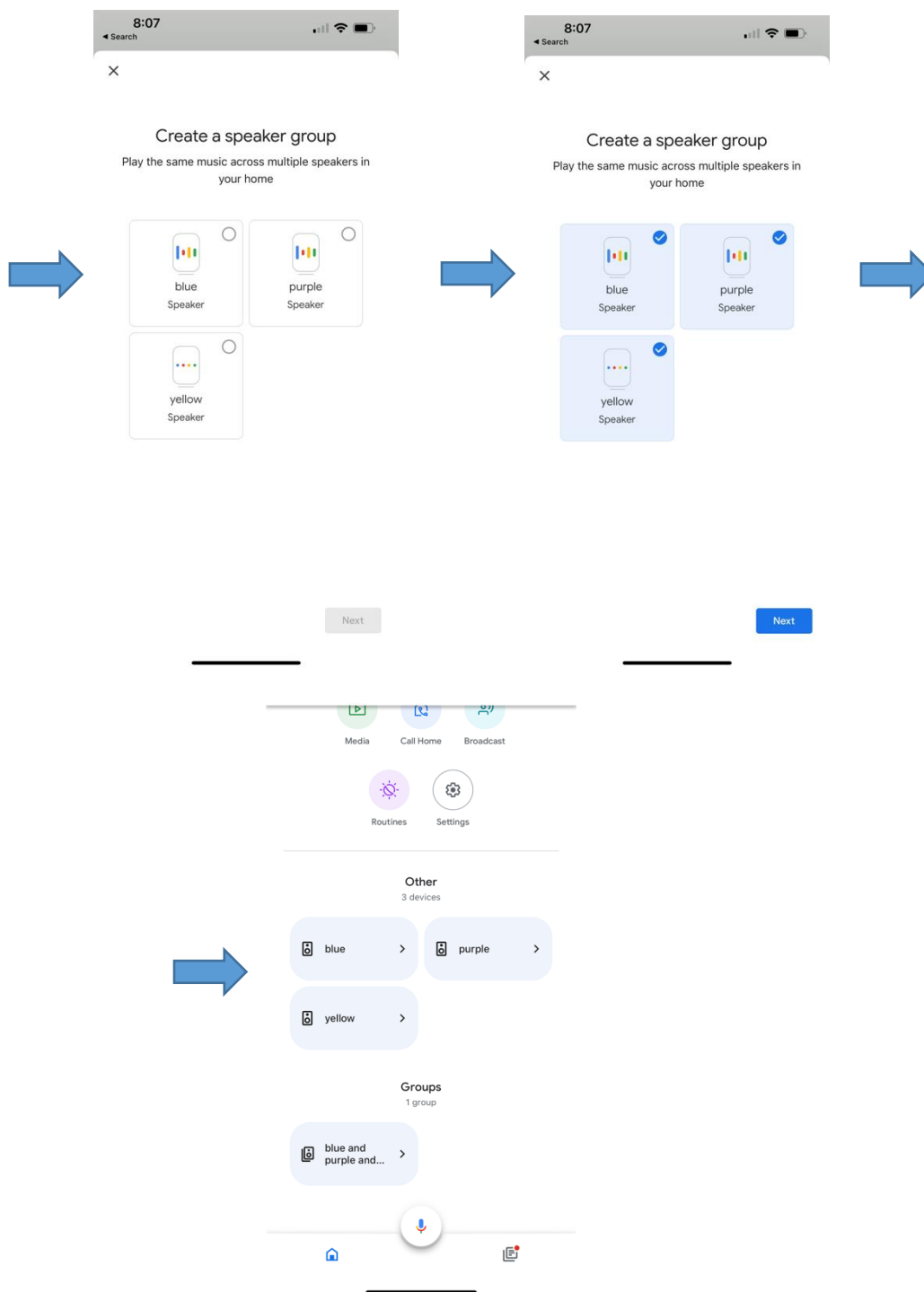
34

ATTORNEYS' EYES ONLY – SOURCE CODE

while no music is playing back to the speaker named "yellow." The Google Home app shows this activity and a group is created including the speakers named "blue," "purple," and "yellow." Immediately after creating the group, the group of speakers outputs no music. As part of that process, both speakers that were playing music are stopped and operate as a group playing back no music.

35

ATTORNEYS' EYES ONLY – SOURCE CODE

58.   As shown above, in every instance, the speakers added to the group act in conjunction with the group once the new group is created. The speakers leave their prior state, stop playback in unison, and remain stopped in conjunction with the group.

106.   This process of leaving the "standalone mode" takes place immediately, and therefore the accused "zone player" does not operate in the alleged "standalone mode" after receiving a first request and will not be operating in the alleged "standalone mode" upon receiving a second request.   Accordingly, any such speaker does not operate in the accused "standalone mode" as claimed.

107.   Furthermore, for the same reasons, to the extent that an accused "zone player" can or does "operat[e] in a standalone mode in which the first zone player is configured to play back media individually," even when "receiving a [first/second] request to create a [first/second] zone scene comprising a [first/second] predefined grouping of zone players including at least the first zone player and a [second/third] zone player," as stated above, any speaker added to a speaker group immediately begins operating as a member of the group and thus immediately begins to play or not play music in synchrony, based on the current operation of the group, as soon as the speaker has been added to the group—*before* the user invokes the modified speaker group. Therefore, the zone scene is not "configured for synchronous playback of media *when the* [*first/second*] *zone scene is invoked*." The requirement that the first zone scene continues to operate in the standalone mode until "receiving a third request to invoke the first zone scene," based on which "the first zone player transition[s] from operating in the standalone mode" is further clarified by Limitations [1.10]-[1.11] / [9.7]-[9.8] as discussed below. Accordingly, any such speaker does not operate in the accused "standalone mode" as claimed.

108.   As a further example, Limitations [1.6] and [1.8] / [9.3] and [9.5] require the "controller" to take particular actions based on the first and second requests, namely to "i) caus[e] creation of the first[/second] zone scene, ii) caus[e] an indication of the first[/second] zone scene to be transmitted to the first[/second] zone player, and iii) caus[e] storage of the first[/second] zone

scene." As I described above, however, the accused steps do not take place while the accused first "zone player" is in a standalone mode. Rather, as I have described, immediately upon making the request to add a speaker to a group, the speaker being added to the group or the group being created begins playback or non-playback as a group, rather than remaining in the accused "standalone mode." As one specific example, ███████████████████████████████

██   ████████████████

██

█████████████████████████ ("StopCurrentApp"). And only *after* this playback is terminated on the speaker is the new group then added through the AddGroup function. Accordingly, the "controller" does not "i) caus[e] creation of the first[/second] zone scene, ii) caus[e] an indication of the first[/second] zone scene to be transmitted to the first[/second] zone player, and iii) caus[e] storage of the first[/second] zone scene" while the accused "zone player" remains in the accused "standalone mode."

109.   Additionally, as I discussed above in Section IX, even if group information was stored prior to "StopCurrentApp()" - and it is not - any leader election for the group leader of the updated group would only take place *after* the AddGroup(g) command.

110.   ***Claims 1 and 9 make clear that steps [1.5]-[1.10] / [9.2]-[9.7] take place while the accused first zone player is (and remains) in standalone mode.***

111.   Dr. Almeroth responds that claim Limitations [1.5]-[1.10] need not take place while any "zone player" is in standalone mode provided in Limitation [1.4]:

> In my opinion, claim limitations 1.5-1.10/9.2-9.7 of claims 1 and 9 of the '966 patent can be met by a "computing device" that is capable of performing each respective function of claim limitations 1.5-1.10/9.2-9.7 at a time when the "first zone player" is operating in "standalone mode," regardless of whether the "first zone player" "continues to operate in "standalone mode" during the time in-between when any two of the respective functions recited in claim limitations

DATED: January 13, 2023

____

Dan Schonfeld, Ph.D.